

2. Digital playgrounds for early computing / D. Kumar // ACM Inroads. 2014. № 5. С. 20-21.
3. An automated learning system for Java programming / C. Daly, J.M. Horgan // IEEE Transactions on Education. 2004. № 47. С. 10-17.
4. Designing Tangible Programming Languages for Classroom Use / M.S. Horn, R. Jacob // Proceedings of the 1st international conference on Tangible and embedded interaction. 2007. С. 159-162.
5. Comparing the Use of Tangible and Graphical Programming Languages for Informal Science Education / M.S. Horn, E.T. Solovey, R.J. Crouser, R. Jacob. // Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. 2009. С. 975-984.
6. Integrating New Technologies and Existing Tools to Promote Programming Learning / Á. Santos, A. Gomes, A.J. Mendes // Algorithms. 2010. № 3. С. 183-196.
7. A Survey of Literature on the Teaching of Introductory Programming / A. Pears, S. Seidman, L. Malmi // ACM SIGCSE Bulletin. 2007. № 39. С. 204-223.
8. Model-driven processes and tools to design robot-based generative learning objects for computer science education / V. Štuikys, R. Burbaitė, K. Bespalova, G. Ziberkas // Science of Computer Programming. 2015. № 129. С. 48-71.
9. NoobLab: An E-learning Platform for Teaching Programming / P. Neve, G. Hunter, D. Livingstone // IEEE 8th International Symposium on Embedded Multicore/Manycore SoCs. 2014.
10. RoboBlock: A remote lab for robotics and visual programming / I. Angulo, J. García-Zubía, U. Hernández-Jayo // 4th Experiment@International Conference. 2017.
11. Engaging students in programming / M. Corney, D. Teague, R.N. Thomas // Proceedings of the Twelfth Australasian Conference on Computing Education. 2010. № 103. С. 63-72.
12. What drives a successful e-Learning? An empirical investigation of the critical factors influencing learner satisfaction / P. Sun, R. J. Tsai, G. Finger // Computers & Education. 2008. № 50. Pp. 1183-1202.
13. Scratch Programming Language // MIT Media Lab, USA. 2002.

УДК 007.51

**А. А. Набатова, В. А. Гольцев**

ФГАОУ ВО «Уральский федеральный университет

имени первого Президента России Б.Н. Ельцина», г. Екатеринбург, Россия

## **ОЦЕНКА ПРОГРАММНЫХ ПРОДУКТОВ ПО РАЗРАБОТКЕ ИНТЕРАКТИВНЫХ ПРИЛОЖЕНИЙ В WINDOWS**

### **Аннотация**

*Целью работы является выбор программного продукта для разработки интерактивного приложения при создании виртуального лабораторного практикума. Для*

создания интерактивного приложения существует ряд программ как с платным, так с бесплатным пакетом. В результате анализа программных продуктов для написания виртуального практикума была выбрана среда разработки Unity 3D. Данная среда разработки имеет удобный интерфейс, а также большое количество обучающего материала, который есть в открытом доступе на официальном сайте. Unity 3D имеет бесплатный контент, который не ограничен в функциях.

**Ключевые слова:** Unity 3D, виртуальная лаборатория, интерактивное приложение, среда разработки.

### **Abstract**

*The aim of the work is to select a software product for developing an interactive application when creating a virtual laboratory practical work. To create an interactive application, there are a number of programs with both paid and free packages. As a result of the analysis of software for writing a virtual workshop, the Unity 3D development environment was chosen. This development environment has a user-friendly interface, as well as a large amount of educational material that is publicly available on the official website. Unity 3D has free content that is not limited in features.*

**Key words:** Unity 3D, virtual lab, interactive application, development environment.

В настоящее время подготовка грамотных специалистов невозможна без применения новых форм обучения. Одной из которой являются использование интерактивных приложений. Интерактивное приложение – это приложения, которые взаимодействуют с пользователем.

Использование интерактивных приложений в качестве виртуальных лабораторных работы позволяет студентам более тщательно подготовиться к экспериментам на реальных физических установках, приобрести навыки работы с измерительными приборами, вникнуть в суть исследуемых процессов.

Для создания интерактивного приложения существует ряд программ как с платным, так с бесплатным пакетом. Написать интерактивное приложение уже не составляет таких затрат как раньше, но выбор программного продукта, на котором следует создать данное приложение, потребует немало времени. Нашей задачей стоит написание продукта для виртуального практикума. Опираясь на это и будем подбирать продукт для разработки.

Рассмотрим следующие программные продукты для разработки виртуального практикума [1]:

Blender – бесплатный пакет профессиональных программ для создания интерактивных приложений и трехмерной компьютерной графики. Инструментов для работы хватит как новичкам, так и профессионалам. Blender содержит в себе средства для моделирования, анимации, обработки видео и звука. Программа представляет собой полноценный редактор, в который уже помещены основные текстуры, обработчики событий и модели. Для получения дополнительных возможностей можно скачать плагины: их создают как официальные разработчики, так и пользователи.

Unity 3D – это мощная среда для разработки приложений и игр, в том числе для мобильных устройств. Созданные на Unity 3D приложения работают на Windows, iOS, Android, Playstation 3, Xbox 360 и Wii. Скрипты в основном создаются на JavaScript, но код можно писать и на C#. Так же обучающие материалы по работе в среде можно найти на официальном сайте [2].

Construct Classic – конструктор двумерных и трехмерных приложений с открытым исходным кодом. Для работы не нужны знания программирования. Достаточно добавить объект и включить анимацию. Но, к сожалению, нет русскоязычной версии. Интерфейс очень понятный, так что можно работать даже с базовыми знаниями английского. Конструктор не только бесплатный – у него открытый исходный код, и при желании его можно настроить и отредактировать.

Game Maker Lite – бесплатная программа для разработки интерактивных приложений. Отличается не сложным интерфейсом и простым программированием. Можно использовать собственные изображения и эффекты, или встроенные программы. Чтобы получить доступ к большому выбору изображений и эффектов, нужно зарегистрироваться. Для работы не нужны знания программирования, но некоторые скрипты можно прописывать и самостоятельно.

Unreal Development Kit – бесплатный движок для создания игр. Очень мощный, с массой возможностей и инструментов для продвинутых визуализаций, и детальных симуляций. Можно создавать интерактивные приложения для множества современных платформ. В состав программы уже входят текстуры, модели, звуки, спрайты, скрипты. Остается только комбинировать и создавать свое приложение.

Game Editor – редактор для создания простых двухмерных игр и приложений, под операционные системы Windows, iOS, Android, Linux. В приложении есть встроенные наборы анимаций, который отвечают за внешний вид созданных объектов. Можно использовать свои графические элементы. Также программа предоставляет стандартный набор реакций, которые определяют поведение объекта в интерактивном приложении. Можно создавать и собственное поведение объекта, на специальном скриптовом языке Game Editor.

3D Rad – Бесплатная программа для разработки 3D-игр и интерактивных приложений. Использовать код не нужно, поэтому создавать свои приложения достаточно просто. Приложение создается путем выбора различных объектов и настройки взаимодействия между ними. Есть функция импорта моделей, большое количество примеров и образцов. Также есть возможность встраивать приложение и игры на веб-страницы.

NeoAxis 3D Engine – Универсальная среда для разработки 3D проектов. Это готовый движок со своими моделями, текстурами, физикой, шаблонами и графикой. На нем можно создавать не только приложения, но и одиночные модели, сложную визуализацию программного обеспечения.

Adobe Flash – мультимедийная платформа компании Adobe Systems для создания веб-приложений, интерактивных приложений или мультимедийных презентаций. Широко используется для создания анимации, игр, а также воспроизведения на веб-страницах видео- и аудиозаписей. Платформа включает в себя ряд средств разработки, прежде всего Adobe Animate и Adobe Flash Builder. Данная программа позволяет работать без особых навыков программирования, но для создания уникального продукта следует использовать язык программирования Actionscript 2.0 или Actionscript 3.0

Программный продукт Blender, Unreal Development Kit не подходит нам по завышенным системным требованиям.

Construct Classic не имеет поддержки русского языка поэтому на ее изучения понадобится больше времени.

В Game Maker Lite крайне неудобное 3D моделирование, что играет главную роль в нашем проекте.

Программный продукт Game Editor может работать только с двумерной графикой, а нас интересует именно трехмерная, для более полного погружения студента в виртуальный мир.

В связи с тем, что на 3D Rad мало обучающего материала, возникнут сложности с его изучением.

Таким образом для реализации нашего проекта будет выбрана программа Unity 3D.

Более подробно рассмотрим программный продукт Unity 3D [2]. Редактор Unity имеет простой интерфейс, который легко настраивать, состоящий из различных окон, благодаря чему можно производить отладку игры прямо в редакторе. Движок поддерживает два скриптовых языка: C#, JavaScript (модификация). Ранее была поддержка Boo (диалект Python), но его убрали в 5-й версии. Расчёты физики производит физический движок PhysX от NVIDIA.

Проект в Unity делится на сцены – отдельные файлы, содержащие свои игровые миры со своим набором объектов, сценариев, и настроек. Сцены могут содержать в себе как, собственно, объекты (модели), так и пустые игровые объекты – объекты, которые не имеют модели («пустышки»). Объекты, в свою очередь содержат наборы компонентов, с которыми и взаимодействуют скрипты. Также у объектов есть название (в Unity допускается наличие двух и более объектов с одинаковыми названиями), может быть тег (метка) и слой, на котором он должен отображаться. Так, у любого объекта на сцене обязательно присутствует компонент Transform — он хранит в себе координаты местоположения, поворота и размеров объекта по всем трём осям. У объектов с видимой геометрией также по умолчанию присутствует компонент Mesh Renderer, делающий модель объекта видимой. Также Unity поддерживает физику твёрдых тел и ткани, а также физику типа Ragdoll (тряпичная кукла). В редакторе имеется система наследования объектов; дочерние объекты будут повторять все изменения позиции, поворота и масштаба родительского объекта. Скрипты в редакторе прикрепляются к объектам в виде отдельных компонентов. Редактор Unity имеет компонент для создания анимации, но также анимацию можно создать предварительно в 3D-редакторе и импортировать вместе с моделью, а затем разбить на файлы. [2]

Unity 3D поддерживает систему Level Of Detail (сокр. LOD), суть которой заключается в том, что на дальнем расстоянии от объекта высокодетализированные модели заменяются на менее детализированные, и наоборот, а также систему Occlusion culling, суть которой в том, что у объектов, не попадающих в поле зрения камеры не визуализируется геометрия и коллизия, что снижает нагрузку на центральный процессор и позволяет оптимизировать проект. При компиляции проекта создается исполняемый (.exe) файл

приложения (для Windows), а в отдельной папке — данные игры (включая все игровые уровни и динамически подключаемые библиотеки) [2].

Движок поддерживает множество популярных форматов. Модели, звуки, текстуры, материалы, скрипты можно запаковывать в формат .unityassets и передавать другим разработчикам, или выкладывать в свободный доступ. Этот же формат используется во внутреннем магазине Unity Asset Store, в котором разработчики могут бесплатно и за деньги выкладывать в общий доступ различные элементы, нужные при создании игр или виртуальных практикумов. Чтобы использовать Unity Asset Store, необходимо иметь аккаунт разработчика Unity. Unity имеет все нужные компоненты для создания мультиплеера. Также можно использовать подходящий пользователю способ контроля версий. К примеру, Tortoise SVN или Source Gear [3].

В Unity входит Unity Asset Server — инструментарий для совместной разработки на базе Unity, являющийся дополнением, добавляющим контроль версий и ряд других серверных решений [2].

Системные требования для Windows зависят, главным образом, от сложности проекта.

С одной стороны, существующие виртуальные лаборатории для обучения металлургическим процессам должны быть улучшены из-за нескольких проблем, таких как 2D-форма учебной среды, слабая привлекательность для молодых студентов и слабая реальность экспериментов. Поэтому разработка лабораторных работ в 3D будет более интересна и привлекательна для студентов.

Трехмерная графика — это целая наука, область, в которой можно совершенствовать свои знания и умения на протяжении всей жизни. Тем не менее для каждого ПО характерен свой специфический набор средств, определяющий область, в которой 3D-редактор удобно применять. Редактор Unity имеет простой интерфейс, который легко настраивать, состоящий из различных окон, благодаря чему можно производить отладку игры прямо в редакторе. Движок поддерживает два скриптовых языка: C#, JavaScript (модификация) [3].

Принято решение разработать виртуальный лабораторный практикум на основании среды разработки Unity 3D.

### **Список использованных источников**

1. Арстанова Л.Г. Занятия и развлечения со старшими дошкольниками. Разработки занятий, бесед, игр и развлечений на нравственные темы / Л.Г. Арстанова. – М.: Учитель, 2017. – 324 с.
2. Unity в действии. Мультиплатформенная разработка на C#. – СПб.: Питер, 2018. – 608 с.
3. Вакуленко Ю.А. Веселая грамматика. Разработки занятий, задания, игры / Ю.А. Вакуленко. – М.: Учитель, 2017. – 780 с.